PROBLEM. Food (contributed by N. R. Lim-Cheng) [Score: 50/50]

Many people are conscious with what they eat nowadays. Many count calories as they strive to maintain or target a certain weight range. Some have restrictions in their diet, due to health conditions or could be by choice. Thus, some go for more vegetables, some would go for more protein, etc.

Refer to the following sample data: *(Source: https://www.calories.info/)*

foodList (with nElem = 6):

foodName: carrot category: vegetable calories: 0.41	egg protein 0.97	pork protein 1.96	cabbage vegetable 0.25	tofu protein 0.76	celery vegetable 0.15	…

menu (with nMenu = 4)

viandName: adobo aList	lumpia	tofuSisig	chopsuey	…
foodName: pork quantity: 200	carrot 100	tofu 300	cabbage 150	
chicken 200	tofu 100	: : :	carrot 150	
egg 120	cabbage 100		celery 100	
: : :	pork 25		pork 50	
	: : :		: : :	
numIng: 3 totalCal: ? (508.4)	4 ? (191.00)	1 ? (228.0)	4 ? (212.00)	

For this problem, you need to implement the functions for each of the following features:

1.) Determine number of calories (per gram) of a certain foodName [10 pts]
Given the data in foodList above, with nElem = 6:
   a.) Determining number of calories of "cabbage" would result to function returning 0.25.
   b.) Determining number of calories of "beef" would result to function returning 0.00.

2.) Determine number of food are of a certain category [10 pts]
   a.) Determining number of food that are "vegetable" in menu[1] will result to function returning 2
   b.) Determining number of food that are "nuts" in menu[1] will result to function returning 0 (none of the ingredients has category nuts)
   c.) Determining number of food that are "protein" in menu[0] will result to function returning 2 (chicken is not found)

3.) Determine total number of calories a certain viand [15 pts]
   a.) Determining total calories of menu[1] will result to totalCal member of menu[1] to be updated to 191.00
   b.) Determining total calories of menu[0] will result to totalCal member of menu[0] to be updated to 508.4 (chicken is not found and is thus 0)

4.) Generate a list of recommended menu items that do not contain certain restricted food (compared against the food name, not the category) and is predominantly containing food of a certain category [15 pts]
   a.) If there is restriction on "egg" and at least 70% of number of ingredients should be "protein", *pReco should be updated to 1 and recommended should contain the following:

tofuSisig tofu 300	…	…

: : : 1 ?		

b.) If there is restriction on “none” and at least 70% of number of ingredients should be “protein”, *pReco should be updated to 1 and recommended should contain the following: (adobo will result to 66.67% protein only because chicken is not found – 2 out of 3 are protein)

tofuSisig tofu 300 : : : 1 ?	…	…

c.) If there is restriction on “fish” and at least 70% of number of ingredients should be “vegetable”, *pReco should be updated to 1 and recommended should contain the following:

chopsuey cabbage 150 carrot 150 celery 100 pork 50 : : : 4 ? (212.00)	…	…

d.) **IF the category of tofu in foodList was vegetable** (instead of protein). Then, If there is restriction on “nuts” and at least 70% of number of ingredients should be “vegetable”, *pReco should be updated to 3 and recommended should contain the following:

lumpia carrot 100 tofu 100 cabbage 100 pork 25 : : : 4 ? (166.25)	tofuSisig tofu 300 : : : 1 ? (228.0)	chopsuey cabbage 150 carrot 150 celery 100 pork 50 : : : 4 ? (212.00)	…

General Instructions and Restrictions:

1. Use and follow the given file templates/skeleton files.
2. Your solution should NOT include any input or output statements (e.g., scanf() or printf())
3. You are only allowed to use strlen(), strcmp(), strcat(), and strcpy() from string.h. No other functions from string.h can be used.
4. You are not allowed to include other header files in the templates, other than those indicated.
5. Your submitted file LASTNAME-food.c should not contain main().

You are given four files for this problem:

(1) `food.h` which contains the type declaration and function prototypes that will be used in the program. This file cannot be modified.

(2) `LASTNAME-FOOD.c` which contains some initial code that you'll need to complete. Comments indicate the requirement and restrictions imposed for the solution. For this file, the student's implementation should not include any scanf) or printf(). This file should not be modified to contain the main().

(3) `foodMain.c` which contains the main() that can be used to test the requirement

(4) `EXPECTED.txt` contains the expected output of running the executable file generated from foodMain.c. For ease in debugging and testing, it is recommended that the student use output redirection. Assuming you have a.exe as the executable file from **compiling foodMain.c,** to do input and output redirection, type the following in the command prompt :
For Windows: `a > LASTNAME-ACTUAL.txt`

You can then open LASTNAME-ACTUAL.txt and compare the contents to EXPECTED.txt. Note that even the spacing should be exactly the same.

DELIVERABLE: Your C program source file `LASTNAME-FOOD.c` with your own last name as filename. For example, if your lastname is SANTOS, then the source file should be named as `SANTOS- FOOD.c`. Upload your source file in AnimoSpace before the indicated submission time.

TESTING & SCORING:

- Your program will be compiled via gcc -Wall, using C99 standard. Thus, for each function that does not compile successfully, the score for that function is 0.
- Your program will be tested by your instructor with other TEXT FILES (contents are different from the ones given to you) and with function calls of different parameter values.
- Full credit will be given for the function only if the student's implementation are all correct for all the test values used by the instructor during checking AND only if the student's implementation complied with the requirement and did not violate restrictions. Deductions will be given if not all test cases produce correct results OR if restrictions were not followed.